

**Exemplul 5.1**

```

BEGIN
    DELETE FROM tip_plata
    WHERE id_tip_plata NOT IN
        (SELECT id_tip_plata FROM facturi);

    -- cursor deschis?
    IF SQL%ISOPEN
    THEN
        DBMS_OUTPUT.PUT_LINE('Cursor deschis');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Cursor inchis');
    END IF;

    -- a gasit linii?
    IF SQL%FOUND
    THEN
        DBMS_OUTPUT.PUT_LINE('A fost gasita cel putin o linie');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Nu a fost gasita nicio linie');
    END IF;

    -- cate linii a gasit
    DBMS_OUTPUT.PUT_LINE('Au fost sterse ' || SQL%ROWCOUNT
        || ' linii');

END;
/

```

**Exemplul 5.4**

```

DECLARE
    CURSOR c IS
        SELECT id_categorie, denumire
        FROM categorii
        WHERE id_parinte IS NULL;

    v_id_categorie categorii.id_categorie%TYPE;
    v_denumire     categorii.denumire%TYPE;

BEGIN
    OPEN c;
    LOOP
        FETCH c INTO v_id_categorie, v_denumire ;
        EXIT WHEN c%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_id_categorie || ' ' || v_denumire);
    END LOOP;
    CLOSE c;
END;
/

```

**Exemplul 5.5**

```

DECLARE
  CURSOR c IS
    SELECT *
    FROM categorii
    WHERE id_parinte IS NULL;

    v_categorii categorii%ROWTYPE;

BEGIN
  OPEN c;
  FETCH c INTO v_categorii;
  WHILE c%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_categorii.id_categorie || ' ' ||
                          v_categorii.denumire);
    FETCH c INTO v_categorii;
  END LOOP;
  CLOSE c;
END;
/

```

**Exemplul 5.6**

```

DECLARE
  TYPE tab_imb IS TABLE OF categorii%ROWTYPE;
  v_categorii tab_imb;

  CURSOR c IS
    SELECT * FROM categorii
    WHERE id_parinte IS NULL;

BEGIN
  OPEN c;
  FETCH c BULK COLLECT INTO v_categorii;
  CLOSE c;
  FOR i IN 1..v_categorii.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(v_categorii(i).id_categorie || ' ' ||
                          v_categorii(i).denumire);
  END LOOP;
END;
/

```

**Exemplul 5.7**

```
--limitarea numarului de linii incarcate

DECLARE
    TYPE tab_imb IS TABLE OF produse.denumire%TYPE;
    v_produse tab_imb;
    v_denumire produse.denumire%TYPE;

    CURSOR c1 IS
        SELECT denumire
        FROM produse
        WHERE ROWNUM <=10;

    CURSOR c2 IS
        SELECT denumire
        FROM produse;

BEGIN
    OPEN c1;
    LOOP
        FETCH c1 INTO v_denumire;
        EXIT WHEN c1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_denumire);
    END LOOP;
    CLOSE c1;

    DBMS_OUTPUT.PUT_LINE ('-----');

    OPEN c2;
    FETCH c2 BULK COLLECT INTO v_produse LIMIT 10;
    CLOSE c2;
    FOR i IN 1..v_produse.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(v_produse(i));
    END LOOP;
END;
/
```

**Exemplul 5.8**

```
DECLARE
    CURSOR c IS
        SELECT *
        FROM categorii
        WHERE id_parinte IS NULL;

BEGIN
    FOR i IN c LOOP
        DBMS_OUTPUT.PUT_LINE(i.id_categorie || ' '
            || i.denumire);
    END LOOP;
END;
/
```

**Exemplul 5.9**

```
BEGIN
  FOR i IN (SELECT *
              FROM categorii
              WHERE id_parinte IS NULL) LOOP
    DBMS_OUTPUT.PUT_LINE(i.id_categorie || ' '
                         || i.denumire);
  END LOOP;
END;
/
```

**Exemplul 5.10**

```
DECLARE
  CURSOR categ IS SELECT id_categorie, denumire
                  FROM categorii
                  WHERE nivel = 5;
  CURSOR prod(v_categ categorii.id_categorie%TYPE) IS
    SELECT MAX(p.denumire), SUM(cantitate)
      FROM produse p, facturi_produse fp
     WHERE v_categ = p.id_categorie
       AND p.id_produs = fp.id_produs
    GROUP BY p.id_produs
   ORDER BY 1,2 desc;

  c_denumire categorii.denumire%TYPE;
  c_id categorii.id_categorie%TYPE;
  p_denumire produse.denumire%TYPE;
  p_cantitate NUMBER;

BEGIN
  OPEN categ;
  LOOP
    FETCH categ INTO c_id, c_denumire;
    EXIT WHEN categ%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(c_denumire);
    DBMS_OUTPUT.PUT_LINE('-----');

    OPEN prod(c_id);

    LOOP
      FETCH prod INTO p_denumire, p_cantitate;
      EXIT WHEN prod%NOTFOUND OR prod%ROWCOUNT>3;
      DBMS_OUTPUT.PUT_LINE(prod%ROWCOUNT || '. ' ||
                           p_denumire || ' ' || p_cantitate);
    END LOOP;

    IF prod%ROWCOUNT = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Nu are produse vandute!');
    END IF;

    CLOSE prod;
    DBMS_OUTPUT.NEW_LINE;
  END LOOP;
  CLOSE categ;
END;
/
```

**Exemplul 5.12**

```

DECLARE
  CURSOR c IS
    SELECT id_produs
    FROM produse
    WHERE id_categorie IN
      (SELECT id_categorie
       FROM categorii
       WHERE denumire = 'Placi de retea Wireless')
    FOR UPDATE OF pret_unitar NOWAIT;

BEGIN
  FOR i IN c LOOP
    UPDATE produse
    SET pret_unitar = pret_unitar*0.95
    WHERE CURRENT OF c;
  END LOOP;
  -- permanentizare si eliberare blocari
  COMMIT;
END;
/

```

**Exemplul 5.13**

```

-- utilizare ROWID in loc de CURRENT OF

DECLARE
  CURSOR c IS
    SELECT id_produs, rowid
    FROM produse
    WHERE id_categorie IN
      (SELECT id_categorie
       FROM categorii
       WHERE denumire = 'Placi de retea Wireless')
    FOR UPDATE OF pret_unitar NOWAIT;

BEGIN
  FOR i IN c LOOP
    UPDATE produse
    SET pret_unitar = pret_unitar*0.95
    WHERE ROWID = i.ROWID;
  END LOOP;
  COMMIT;
END;
/

```

**Exemplul 5.14**

```

DECLARE
    TYPE tip_cursor IS REF CURSOR RETURN produse%ROWTYPE;
    c tip_cursor;

    v_optiune NUMBER(1) := &p_optiune;
    i produse%ROWTYPE;

BEGIN
    IF v_optiune = 1 THEN
        OPEN c FOR
            SELECT *
            FROM produse p
            WHERE EXISTS (SELECT 1
                            FROM facturi_produse pf, facturi f
                            WHERE p.id_produs = pf.id_produs
                            AND pf.id_factura = f.id_factura
                            AND TO_CHAR(data,'q') = 1);

    ELSIF v_optiune = 2 THEN
        OPEN c FOR
            SELECT *
            FROM produse p
            WHERE id_produs IN
                (SELECT id_produs
                    FROM facturi_produse pf, facturi f
                    WHERE pf.id_factura = f.id_factura
                    AND TO_CHAR(data,'q') = 2);

    ELSIF v_optiune = 3 THEN
        OPEN c FOR
            SELECT DISTINCT p.*
            FROM produse p, facturi_produse pf, facturi f
            WHERE p.id_produs = pf.id_produs
            AND pf.id_factura = f.id_factura
            AND TO_CHAR(data,'q') = 3;

    ELSE
        OPEN c FOR
            SELECT *
            FROM produse p
            WHERE id_produs IN (SELECT id_produs
                                FROM facturi_produse);

    END IF;

LOOP
    FETCH c INTO i;
    EXIT WHEN c%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(i.id_produs||' '||i.denumire);

END LOOP;
DBMS_OUTPUT.PUT_LINE('Nr produse vandute: ' ||
                     c%ROWCOUNT);
    CLOSE c;
END;
/

```

**Exemplul 5.15**

```
--SQL DINAMIC

DECLARE
    TYPE tip_cursor IS REF CURSOR;
-- ? obtin eroare daca includ RETURN produse%ROWTYPE;
    c tip_cursor;

    v_optiune NUMBER(1) := &p_optiune;
    i produse%ROWTYPE;

BEGIN
    OPEN c FOR
        'SELECT DISTINCT p.*'
        'FROM produse p, facturi_produse pf, facturi f'
        'WHERE p.id_produs = pf.id_produs'
        'AND pf.id_factura = f.id_factura'
        'AND TO_CHAR(data,''q'') = :v'
        'USING v_optiune';
    LOOP
        FETCH c INTO i;
        EXIT WHEN c%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(i.id_produs||' '||i.denumire);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Nr produse vandute: ' ||
                           c%ROWCOUNT);
    CLOSE c;
END;
/

-- ? cum modific astfel incat sa obtin acelasi
-- rezultat ca in exemplul anterior
OPEN c FOR
    'SELECT DISTINCT p.*'
    'FROM produse p, facturi_produse pf, facturi f'
    'WHERE p.id_produs = pf.id_produs'
    'AND pf.id_factura = f.id_factura'
    'AND TO_CHAR(data,''q'') =
        CASE WHEN :v1>3 THEN TO_CHAR(data,''q'')
              ELSE TO_CHAR(:v2)
        END'
    'USING v_optiune, v_optiune;
```

**Exemplul 5.16**

```

DECLARE
  CURSOR categ IS
    SELECT denumire,
      CURSOR (SELECT MAX(denumire)
        FROM produse p, facturi_produse fp
        WHERE c.id_categorie = p.id_categorie
        AND p.id_produs = fp.id_produs
        GROUP BY p.id_produs
        ORDER BY 1, SUM(cantitate) desc)
    FROM categorii c
    WHERE nivel = 5;

  c_denumire categorii.denumire%TYPE;
  v_cursor SYS_REFCURSOR;

  TYPE tab_prod IS TABLE OF produse.denumire%TYPE
    INDEX BY BINARY_INTEGER;
  v_prod tab_prod;

BEGIN
  OPEN categ;

  LOOP
    FETCH categ INTO c_denumire, v_cursor;
    EXIT WHEN categ%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE(c_denumire);
    DBMS_OUTPUT.PUT_LINE('-----');

    FETCH v_cursor BULK COLLECT INTO v_prod LIMIT 3;

    IF v_prod.COUNT = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Nu are produse vandute!');
    ELSE
      FOR i IN v_prod.FIRST..v_prod.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(i || ' . ' || v_prod(i));
      END LOOP;
    END IF;

    DBMS_OUTPUT.NEW_LINE;
  END LOOP;

  CLOSE categ;
END;
/

```